**Spring AOP Overview**

Ví dụ như Logging, transaction management, data validation.

Trong OOP , tính module của ứng dụng đc thực hiện bởi các class trong khi AOP được thực hiện bởi các khía cạnh và chúng đạt được cấu hình để cắt qua các class khác nhau.

Spring AOP đưa ra sự phụ thuộc trực tiếp của các tác vụ crosscutting từ các class mà chúng ta ko thể đạt được thong qua OOP bình thường.

Ví dụ, chúng ta có thể có 1 class riêng biệt để ghi nhật ký, nhưng một lần nữa chức năng sẽ gọi các method này để đạt được việc ghi nhật ký trên ứng dụng.

**Aspect Oriented Programming Core Concepts**

Các khái niệm cốt lõi của AOP

* **Aspect :** Là một class thực hiện các lien quan đã cắt ngang nhiều class, chẳng hạn như transaction. Aspect có thể là một class bình thường được cấu hình thông qua cấu hình Spring XML hoặc chúng ta có thể sử dụng tích hợp Spring AspectJ để định nghĩa một class như Aspect bằng cách sử dụng @Aspect
* **Join Point (Điểm kết nối ) :** Là điểm cụ thể trong ứng dụng như thực thi method, xử lý ngoại lệ, thay đổi giá trị biến đối tượng … Một điểm nối luôn luôn là việc thực hiện 1 method.
* **Advice :** Là các hành động được thực hiện cho một điểm tham gia cụ thể. Về mặt lập trình, chúng là các method được thực thi khi một điểm kết nối nhật định với điểm tương ứng được truy cập trong ứng dụng.
* **PointCut :** Là biểu thức được kết hợp với các join point ( điểm nối ) để xác định xem advice có cần được thực thi hay ko. PointCut được sử dụng các kiểu biểu thức khác nhau được kết hợp với các joinpoint và Spring framework sử dụng ngôn ngữ biểu thức phím tắt AspectJ.
* **Target Object :**